

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-152491

(43)Date of publication of application : 16.06.1995

(51)Int.Cl.

G06F 3/06
G06F 3/08
G11B 20/10

(21)Application number : 05-323176

(71)Applicant : HITACHI LTD

(22)Date of filing : 29.11.1993

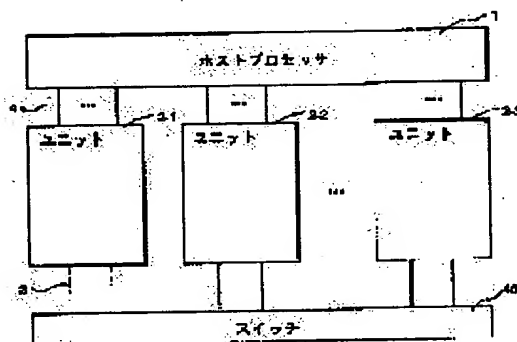
(72)Inventor : KAGIMASA TOYOHICO
TSUNODA HITOSHI
TAKAMOTO YOSHIFUMI

(54) VIRTUAL DISK SYSTEM

(57)Abstract:

PURPOSE: To totally manage a drive in the disk system consisting of many drives by providing plural units with a physical drive and an access control circuit and providing an interface as the virtual drive on a host device.

CONSTITUTION: When an access request comes from a host processor 1 being the host device to a unit 2, each unit 2 converts the logic address of the virtual drive into the physical address of the physical drive and selects the access request to be sent to an access control circuit of the self unit or to the other unit. In short, the physical drive to be accessed is in the self unit, it is sent to the self unit and when it is the other unit, it is sent through a switch 40. Therefore, the virtual drive over all units can be set and an access request from one unit 2 to any area of volume can be sent to an objective unit 2.



LEGAL STATUS

[Date of request for examination] 27.03.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3431972

[Date of registration] 23.05.2003

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's]

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-152491

(43)公開日 平成7年(1995)6月16日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/06	3 0 1 K			
3/08	F			
G 1 1 B 20/10	D	7736-5D		

審査請求 未請求 請求項の数 9 F D (全 16 頁)

(21)出願番号 特願平5-323176

(22)出願日 平成5年(1993)11月29日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 鍵政 豊彦

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 角田 仁

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 高本 良史

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

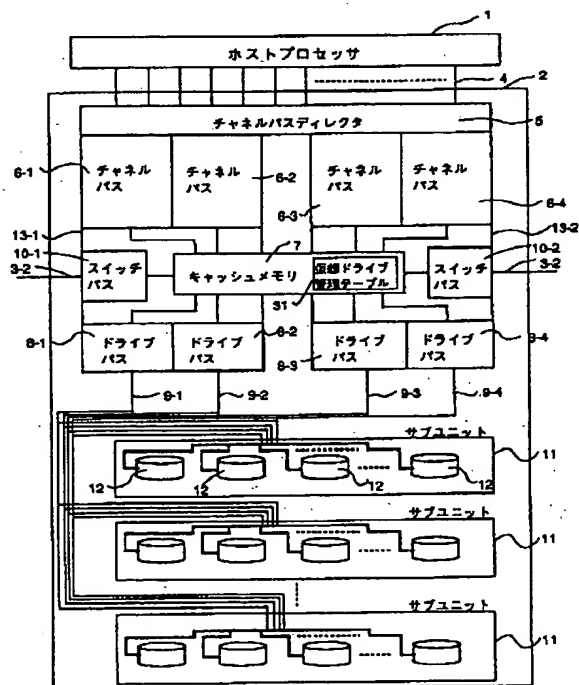
(74)代理人 弁理士 矢島 保夫

(54)【発明の名称】 仮想ディスクシステム

(57)【要約】 (修正有)

【構成】複数の物理ドライブとそれらのアクセス制御回路とを備えたユニットを複数備えるようにし、各ユニットごとに、上位装置から送られる仮想ドライブの論理アドレスを物理ドライブの物理アドレスに変換するとともにその論理アドレスに対応する物理ドライブが自ユニット内にあるか他ユニット内にあるかを判別できる情報を保持するテーブルを備え、そのテーブルを用いて、アクセスすべき物理ドライブが自ユニット内のものであるときは自ユニットで処理し、アクセスすべき物理ドライブが他ユニット内のものであるときはアクセス要求を他ユニットのアクセス制御回路に転送するようにする。

【効果】多数ドライブが実装されていても少数の仮想ディスクボリュームとしてホストプロセッサからアクセスすることにより、統合的な管理ができ、増設が容易となる。



【特許請求の範囲】

【請求項1】複数の物理ドライブと該複数の物理ドライブへのアクセスを制御するアクセス制御回路とを備えたユニットを複数備えるとともに、上位装置には1台の仮想ドライブとしてのインターフェースを提供する仮想ディスクシステムであって、

前記各ユニットは、

前記上位装置から与えられる仮想ドライブの論理アドレスを前記複数の物理ドライブの物理アドレスに変換するアドレス変換手段と、

前記上位装置から仮想ドライブの論理アドレスを指定したアクセス要求があったときは、前記アドレス変換手段により、指定された論理アドレスを前記複数の物理ドライブの物理アドレスに変換するとともに、そのアクセス要求を自ユニットのアクセス制御回路に転送するか、他ユニットのアクセス制御回路に転送するかを選択する手段とを備えたことを特徴とする仮想ディスクシステム。

【請求項2】複数の物理ドライブと該複数の物理ドライブへのアクセスを制御するアクセス制御回路とを備えたユニットを複数備えるとともに、上位装置には1台の仮想ドライブとしてのインターフェースを提供する仮想ディスクシステムであって、

前記各ユニットは、

前記上位装置から与えられる仮想ドライブの論理アドレスを前記複数の物理ドライブの物理アドレスに変換するためのテーブルであって、その論理アドレスに対応する物理ドライブが自ユニット内にあるか他ユニット内にあるかを判別できる情報を保持するテーブルを格納した記憶手段と、

前記上位装置から仮想ドライブの論理アドレスを指定したアクセス要求があったときは、前記テーブルを用いて、指定された論理アドレスを前記複数の物理ドライブの物理アドレスに変換するとともに、その変換結果により、アクセスすべき物理ドライブが自ユニット内のものであるときはアクセス要求を自ユニットのアクセス制御回路に転送し、アクセスすべき物理ドライブが他ユニット内のものであるときはアクセス要求を該他ユニットのアクセス制御回路に転送する手段とを備えたことを特徴とする仮想ディスクシステム。

【請求項3】さらに、複数の前記ユニット間を接続する接続手段を備え、その接続手段を介して前記アクセス要求を他ユニットに転送することを特徴とする請求項1または2に記載の仮想ディスクシステム。

【請求項4】前記テーブルは、仮想ドライブの論理アドレスを保持するフィールドとそれに対応する物理ドライブの物理アドレスを保持するフィールドとを有し、該論理アドレスに対応する物理ドライブが自ユニット内にあるときは、該物理アドレスのフィールドに、該物理ドライブを特定するドライブアドレスとそのドライブ内アドレスとを保持し、該論理アドレスに対応する物理ドライブ

が他ユニット内にあるときは、該物理アドレスのフィールドに、該他ユニットを特定する情報を保持することを特徴とする請求項1から3のいずれか1つに記載の仮想ディスクシステム。

【請求項5】前記ユニットのアクセス制御回路は、前記他ユニットからアクセス要求が転送されてきたときは、該アクセス要求に応じて自ユニット内の物理ドライブをアクセスし、そのアクセス結果を、直接、前記上位装置に伝達することを特徴とする請求項1から4のいずれか1つに記載の仮想ディスクシステム。

【請求項6】前記上位装置から仮想ドライブの作成要求が送られてきたときに、前記テーブルの論理アドレスを保持するフィールドの空きエントリを確保し、作成すべき仮想ドライブの論理アドレスを該フィールドに設定する手段を、さらに備えたことを特徴とする請求項4に記載の仮想ディスクシステム。

【請求項7】前記上位装置から仮想ドライブの領域確保要求が送られてきた場合、領域を確保するユニットとして自ユニットが指定されているときは、自ユニット内の物理ドライブの空き領域を確保して前記テーブルの物理アドレスを保持するフィールドに、確保した空き領域の物理ドライブのドライブアドレスとそのドライブ内アドレスとを格納し、領域を確保するユニットとして他ユニットが指定されているときは、該指定されている他ユニットに領域確保要求を転送する手段を、さらに備えたことを特徴とする請求項6に記載の仮想ディスクシステム。

【請求項8】前記他のユニットから仮想ドライブの領域確保要求が送られてきた場合は、自ユニット内の物理ドライブの空き領域を確保して、前記テーブルの物理アドレスを保持するフィールドに、確保した空き領域の物理ドライブのドライブアドレスとそのドライブ内アドレスとを格納し、領域確保要求の要求元のユニットに伝達を返す手段を、さらに備えたことを特徴とする請求項7に記載の仮想ディスクシステム。

【請求項9】前記ユニット単位で増設を行うことを特徴とする請求項1から8のいずれか1つに記載の仮想ディスクシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ディスクシステムに関し、特に多数のディスクドライブ装置を有する磁気ディスクシステムに適用して好適な仮想ディスクシステムに関する。

【0002】

【従来の技術】現在のコンピュータシステムにおいては、2次記憶装置が用いられ、ホストプロセッサなどの上位装置側が必要とするデータは2次記憶装置に格納される。ホストプロセッサは、必要に応じて、2次記憶装置に対してデータの書込みおよび読み出しを行ってい

る。この2次記憶装置としては、一般に不揮発な記憶媒体が使用され、代表的なものとして磁気ディスク装置、光ディスク装置などがあげられる。磁気ディスク装置は、現在そして近い将来も最も多く使用される2次記憶装置と期待されている。

【0003】ところで、近年のコンピュータシステムの応用の大規模化および多様化にともない、一つのスピンドルにより構成される磁気ディスク装置（以下、ドライブと呼ぶ）では、性能が不足するという状況となってきた。ディスクシステムのデータ転送速度およびトラフィック性能は、スピンドル数すなわちドライブ数に比例して向上する。そのため、多数の小記憶容量のドライブを使用して高性能化を図るアプローチが採られている。従来の大型ディスクシステムではディスク径9インチのドライブを用いていたが、現在ではディスク径2.5インチ程度のドライブが既に製品化され、コンピュータシステムにて使用されている。

【0004】このような多数の小記憶容量のドライブを使用して高性能化を図るアプローチでは、多数ドライブの統合的管理が必要となる。従来のコンピュータシステムでは、1ドライブを1ボリュームという管理単位として、ホストプロセッサのソフトウェアであるオペレーティングシステム（以下、OSと呼ぶ）が管理する。すなわち、ボリュームの空き領域や、格納されているファイルなどに関連する情報は、OSがボリュームごとに管理するのである。

【0005】したがって、ボリュームが多数あると、OSの管理オーバーヘッドが大きいく。それは、OSのファイルシステムの処理時間が長くなるという形で欠点として現われる。また、ボリュームが小容量であると、関連するファイルのグループが一つのボリュームに入りきれないため、論理的には一つのグループのファイルを分割して複数のボリュームに格納せざるを得ず、ユーザにとって使い勝手の悪いディスクシステムとなってしまう。

【0006】そこで、多数のドライブを統合的に管理するとともに、上記の欠点をなくしたOSのボリューム管理方法が、例えば、オレイリー・アソシエイツ社（O'Reilly & Associates, Inc.）刊の「ガイド・トゥー・OSF/1：テクニカル・シノプシス」（1991年）第7-1頁から第7-10頁（Guide to OSF/1: A Technical Synopsis (1991), pp7-1-7-10）に述べられている。以下、この従来技術を第1の従来技術と呼ぶ。

【0007】この第1の従来技術では、論理ボリュームという概念により、OSが複数のドライブを管理し、これにより複数のドライブが一つのボリュームとしてユーザに見えるようにしている。ドライブの記憶領域をエクステンと呼ぶ固定長領域に分割し、一つの論理エクステンに対して一つまたは複数の物理エクステントを対応させる。対応させる物理エクステントは、どの物理ドライブのものでもよく、一つの論理エクステンに異なる

物理ドライブの物理エクステントを対応させてよい。論理エクステントを集めて、一つの仮想的なドライブとしたものをディスクの論理ボリュームと呼ぶ。

【0008】一つの論理エクステンに対して複数の物理エクステントを対応させた場合がミラーリングとなる。ミラーリングは、同じデータを複数の物理記憶装置に格納して、記憶装置の障害時におけるデータの信頼性を高める手法である。

【0009】ファイルに対するread/write時は、OSが、ファイルが格納されている論理ボリュームの管理テーブルを参照してアクセス先の物理ドライブとそのドライブ内の物理アドレスを求め、そこに対してread/writeを発行する。アクセス先がミラーリングされている場合は、対応する複数の物理ドライブのそれぞれに同一データのwriteを発行する。readは、アクセス時間が短い物理ドライブに発行する。

【0010】この第1の従来技術では、物理ドライブを増設した場合に、すでに存在する論理ボリュームの末尾にその物理ドライブのエクステントを追加して、論理ボリュームを拡張することができる。

【0011】複数のドライブにわたってファイルを格納することができる制御装置が、特公平4-30056号の「仮想記憶システム」に開示されている。以下、該公報に記載の従来技術を第2の従来技術と呼ぶ。

【0012】この第2の従来技術では、従来のディスク制御装置とホストプロセッサとの間に、新たな上位の制御装置である仮想制御プロセッサを設ける。仮想制御プロセッサは、ホストプロセッサに対して物理ドライブを見せず、ファイルの1レベルな記憶空間のみを見せる。ホストプロセッサは仮想制御プロセッサに対して、ファイルとファイル内のアドレスを指定してread/writeを発行する。仮想制御プロセッサは、ファイルを記憶装置に適したブロックに分割して、その仮想制御プロセッサに接続されたドライブなどの記憶装置に分散格納する。ファイルのどのブロックがどこに格納されているかは、仮想制御プロセッサ内のテーブルに保持する。

【0013】この第2の従来技術の目的は、本発明とは異なり、記憶領域の効率的使用にある。ブロックごとに自由に格納場所を設定できるので、あらかじめファイル最大容量を確保する従来の方法にくらべ、空き領域のまま確保されている領域が少なくなり、領域を効率的に使用できる。また、制御装置というハードウェアでファイルのブロック単位の分散格納を実現するので、OSが行なうのに比べてOSの処理オーバーヘッドが小さくなるという利点がある。

【0014】

【発明が解決しようとする課題】第1の従来技術では、ホストプロセッサのソフトウェアであるOSが、論理ボリュームを実現している。したがって、OSの論理ボリューム処理に要するCPUオーバーヘッドが大きいという

問題点があった。

【0015】また、複数のホストプロセッサから論理ボリュームを共有することが困難であるという問題点があった。これは、複数のホストプロセッサが別々の論理ボリュームの管理テーブルを保持しているため、複数のホストプロセッサから同一の論理ボリュームとしてアクセスするためには、ホストプロセッサどうしが通信して管理テーブルの整合性を保つ必要があるからである。

【0016】さらに、論理ボリュームを実現するための処理をOSが行っているため、ディスクシステムを改良する場合は、OS自体を変更する必要があるという問題点があった。OSは、コンピュータシステムのメーカーごとに少しずつ異なり、ファイルシステムの改良に応じてそれらのすべてのOSを変更することは現実的でない。

【0017】第2の従来技術では、仮想制御プロセッサが物理的なドライブを管理しているため、ドライブを増設していくと仮想制御プロセッサが性能上のボトルネックとなり、仮想制御プロセッサも増設しなければならない。一つの仮想制御プロセッサは、それに接続されるドライブのみ管理できるが、他の仮想制御プロセッサに接続されるドライブは管理できない。すなわち、統合的に管理可能なドライブの増設は、一つの仮想制御プロセッサに接続する限りにおいてのみ有効であるという問題点があった。具体的には、複数の仮想制御プロセッサにわたってファイルを格納できないということである。

【0018】また、仮想制御プロセッサがホストプロセッサに対して提供するインターフェースは、従来のボリュームという概念ではなく、ファイルを指定してアクセスするという独自のインターフェースであり、ディスクシステムとしての互換性（汎用性）がないという問題点もあった。

【0019】さらに、ホストプロセッサからのアクセス要求は、仮想制御プロセッサおよびディスク制御装置と、二つの制御装置を経由してからドライブに到達する。したがって、アクセス処理時間が長くなるという問題点もあった。

【0020】本発明の目的は、多数のドライブから構成されるディスクシステムの改良にある。また、本発明の目的は、多数のドライブから構成されるディスクシステムにおいて、複数のドライブを統合的に管理でき、増設が容易であり、従来ドライブとの互換性があり、高性能な仮想ディスクシステムを提供することにある。

【0021】

【課題を解決するための手段】上記目的を達成するため、本発明は、複数の物理ドライブと該複数の物理ドライブへのアクセスを制御するアクセス制御回路とを備えたユニットを複数備えるとともに、上位装置には1台の仮想ドライブとしてのインターフェースを提供する仮想ディスクシステムであって、前記各ユニットは、前記上

位装置から与えられる仮想ドライブの論理アドレスを前記複数の物理ドライブの物理アドレスに変換するアドレス変換手段と、前記上位装置から仮想ドライブの論理アドレスを指定したアクセス要求があったときは、前記アドレス変換手段により、指定された論理アドレスを前記複数の物理ドライブの物理アドレスに変換するとともに、そのアクセス要求を自ユニットのアクセス制御回路に転送するか、他ユニットのアクセス制御回路に転送するかを選択する手段とを備えたことを特徴とする。

【0022】また、本発明は、複数の物理ドライブと該複数の物理ドライブへのアクセスを制御するアクセス制御回路とを備えたユニットを複数備えるとともに、上位装置には1台の仮想ドライブとしてのインターフェースを提供する仮想ディスクシステムであって、前記各ユニットは、前記上位装置から与えられる仮想ドライブの論理アドレスを前記複数の物理ドライブの物理アドレスに変換するためのテーブルであって、その論理アドレスに対応する物理ドライブが自ユニット内にあるか他ユニット内にあるかを判別できる情報を保持するテーブルを格納した記憶手段と、前記上位装置から仮想ドライブの論理アドレスを指定したアクセス要求があったときは、前記テーブルを用いて、指定された論理アドレスを前記複数の物理ドライブの物理アドレスに変換するとともに、その変換結果により、アクセスすべき物理ドライブが自ユニット内のものであるときはアクセス要求を自ユニットのアクセス制御回路に転送し、アクセスすべき物理ドライブが他ユニット内のものであるときはアクセス要求を該他ユニットのアクセス制御回路に転送する手段とを備えたことを特徴とする。

【0023】ユニット間のアクセス要求の転送は、ユニット間を接続する接続手段（例えば、ユニット間を直接接続するスイッチパス）によるとよい。ユニットと上位装置との間に設けたチャネルディレクタなどによってもよい。

【0024】前記テーブルは、例えば仮想ドライブの論理アドレスを保持するフィールドとそれに対応する物理ドライブの物理アドレスを保持するフィールドとを有するものである。そして、論理アドレスに対応する物理ドライブが自ユニット内にあるときは、該物理アドレスのフィールドに、該物理ドライブを特定するドライブアドレスとそのドライブ内アドレスとを保持し、該論理アドレスに対応する物理ドライブが他ユニット内にあるときは、該物理アドレスのフィールドに、該他ユニットを特定する情報を保持するようにするとよい。

【0025】前記ユニットのアクセス制御回路は、前記他ユニットからアクセス要求が転送されてきたときは、該アクセス要求に応じて自ユニット内の物理ドライブをアクセスし、そのアクセス結果を、直接、前記上位装置に伝達するようにする。なお、アクセス要求の転送元のユニットに返すようにしてもよいが、直接上位装置に返

すようにしたほうが性能はよい。

【0026】前記上位装置から要求を送って、仮想ドライブの作成や領域確保を行うことができるようにしてもよい。仮想ドライブの作成要求のときは、前記テーブルの論理アドレスを保持するフィールドの空きエントリを確保し、作成すべき仮想ドライブの論理アドレスを該フィールドに設定する。また、仮想ドライブの領域確保要求のときは、領域を確保するユニットとして自ユニットが指定されているときは、自ユニット内の物理ドライブの空き領域を確保して前記テーブルの物理アドレスを保持するフィールドに確保した空き領域の物理ドライブのドライブアドレスとそのドライブ内アドレスとを格納し、領域を確保するユニットとして他ユニットが指定されているときは、該指定されている他ユニットに領域確保要求を転送するようにする。他のユニットから仮想ドライブの領域確保要求が送られてきた場合は、自ユニット内の物理ドライブの空き領域を確保して、前記テーブルの物理アドレスを保持するフィールドに確保した空き領域の物理ドライブのドライブアドレスとそのドライブ内アドレスとを格納し、領域確保要求の要求元のユニットに回答を返すようにする。

【0027】

【作用】上位装置である例えばホストプロセッサからアクセス要求がユニットに到着すると、各ユニットでは、仮想ドライブの論理アドレスを物理ドライブの物理アドレスに変換し、それとともにそのアクセス要求を自ユニットのアクセス制御回路に転送するか、他ユニットのアクセス制御回路に転送するかを選択する。

【0028】すなわち、アクセスすべき物理ドライブが自ユニット内のものであるときはアクセス要求を自ユニットのアクセス制御回路に転送し、アクセスすべき物理ドライブが他ユニット内のものであるときはアクセス要求を該他ユニットのアクセス制御回路に転送する。したがって、ディスクシステムを構成するすべてのドライブにわたる仮想ドライブ（仮想ディスクボリューム）が設定でき、一つのユニットからボリュームのどの領域へのアクセス要求でも対象ユニットに転送可能である。そのため、多数ドライブを統合的に管理でき、増設が容易であり、従来ドライブとの互換性があり、高性能なディスクシステムを実現することができる。

【0029】

【実施例】以下、本発明の一実施例を図1～12を参照して説明する。

【0030】図3は、本発明の一実施例に係る仮想ディスクシステムの全体システム構成を示す。本実施例の仮想ディスクシステムは、複数のユニット2-1、2-2、…、2-3、およびスイッチ40により構成される。ホストプロセッサ1とディスクシステムの各ユニット2とは、チャンネル（外部インターフェースバス）4により接続されている。ホストプロセッサ1には、ディス

クシステムのユニット2を複数接続することができる。

【0031】各ユニット2内には、ディスクドライブが備えられ、データが格納される。スイッチ40は、ユニット2間を接続するためのものである。ホストプロセッサ1は、IBM社のES/9000シリーズ（商品名）のような大型汎用計算機とする。

【0032】オペレーティングシステム（OS）は、ホストプロセッサ1のチャネルインターフェースのコマンド体系にのっとりコマンドを発行する。コマンドで指定されるアドレスは、仮想ドライブ（仮想ディスクボリューム）を特定するドライブアドレス、並びにその仮想ドライブのシリンダアドレス（CC）、およびヘッドアドレス（HH）からなるCCHHである。なお、これにレコードアドレス（R）を加えてCCHHRのアドレスを用いてもよいが、説明を容易にするため、本実施例では、CCHHのアドレスで指定するものとする。

【0033】図2は、仮想ドライブの概念図であり、コマンドで指定するアドレスCCHH（またはCCHHR）を説明するための図である。

【0034】ホストプロセッサ1から見たときの仮想ドライブは、図2のように従来の1つの物理ドライブのように見える。したがって、ホストプロセッサ1からは、まず仮想ドライブをドライブアドレスで特定するとともに、図2に示すように、当該データが格納されている（または格納する）トラックが所属するシリンダの位置を特定するシリンダアドレス（CC）、そのシリンダ内において当該データが格納されている（または格納する）トラックを特定するヘッドアドレス（HH）、および必要ならそのトラック内のレコードの位置を特定するレコードアドレス（R）で、データの位置を特定するようにしている。

【0035】具体的には、ドライブアドレスは要求データが格納されている当該ドライブの番号（ドライブ番号）であり、シリンダアドレス（CC）は当該ドライブ内のシリンダ番号であり、ヘッドアドレス（HH）はそのシリンダにおいてトラックを選択するヘッドの番号であり、レコードアドレス（R）はそのトラック内のデータの位置である。

【0036】従来のCKDフォーマット対応の磁気ディスクサブシステム（IBM3990-3390：商品名）も、図2に示したようなアドレスでアクセスされる。本実施例の仮想ディスクシステムにおいても、同様に、上述したようなアドレスを用いたコマンドがホストプロセッサ1から発行されアクセスされる。

【0037】仮想ドライブは、図2で説明したようなアドレスを持つ仮想的なドライブとして、実際は複数のユニット2の中に作成される。上述したように、一つの仮想ドライブは、ホストプロセッサ1からは一つの物理ドライブに見えるが、その領域は複数のユニット2に分散していてもかまわない。

【0038】仮想ドライブの領域は、トラック単位で物理ドライブの領域に分散される。すなわち、仮想ドライブの一つのトラックは、ユニット2内の物理ドライブの一つのトラックに対応づけられる。それは、主記憶装置における仮想記憶ページと実記憶ページとの関係と同様である。

【0039】ホストプロセッサ1から上述したようなアドレスを用いて、リード要求やライト要求などのコマンドが発行されたとき、ユニット2は、自ユニット内に要求されたアドレスに対応する物理ドライブがあるときはその物理ドライブにアクセスして結果をホストプロセッサ1に返す。また、ユニット2は、指定されたアドレスが自ユニット内でなく他ユニット内の物理ドライブに対応するときは、スイッチ40を介して他ユニットにリードやライトの要求を発行（転送）する。そのような要求を受けたユニット2では、その要求に応じて、ユニット内の物理ドライブにアクセスし、結果を直接ホストプロセッサ1に返す。

【0040】大型計算機においては、ホストプロセッサ1が要求を発行したチャンネルとその要求に対する応答の割り込みが返ってくるチャンネルとは、異なっているかもしれない。入出力デバイスは、任意の空きのチャンネルを確保して応答を返す。本実施例のシステムにおいても、ホストプロセッサ1は、ユニット2の応答がどのチャンネル4から返ってきてもよい構成となっている。したがって、例えば図3において、ユニット2-1に発行したリード要求の応答が、ユニット2-2に接続されたチャンネル4から返ってきることができる。そうした方が、応答のユニット間転送が無くなり、より高性能となる。

【0041】本実施例において、ユニット2を増設するときは、増設ユニット2をホストプロセッサ1とスイッチ40に接続する。接続した後に、後で説明する仮想ドライブ作成コマンドや領域確保コマンドを発行して増設ユニット2を仮想ドライブとして組み込む。

【0042】本実施例において、ホストプロセッサ1から仮想ディスクシステムに発行するコマンドの種類としては、ディスクのデータをホストプロセッサ1に読み込むリードコマンド、およびデータをディスクに書き込むライトコマンドがある。これらは、従来からあるコマンドであり、本実施例においても重要なコマンドであるので後に詳しく説明する。その他の従来のコマンドは、本発明に直接影響しないのでここでは説明しない。

【0043】本実施例では、ホストプロセッサ1から発行するコマンドとして、仮想ドライブ作成コマンドと領域確保コマンドとを新設する。仮想ドライブ作成コマンドは、新たな仮想ディスクドライブをディスクシステム内に作成するコマンドである。仮想ドライブ作成コマンドを発行するときは、仮想ドライブの管理元のユニット番号と仮想ドライブアドレスとをオペランドとして指定する。領域確保コマンドは、仮想ドライブに新たな領域

を確保するコマンドである。領域確保コマンドを発行するときは、仮想ドライブアドレスと実際の領域を確保するユニットのユニット番号と物理ドライブアドレスと確保トラック数とをオペランドとして指定する。

【0044】上記の仮想ドライブ作成コマンドおよび領域確保コマンドに対するディスクシステムからのホストプロセッサ1への応答としては、それぞれ、成功または失敗を知らせる割り込みがある。

【0045】ホストプロセッサ1から発行されるコマンドのほかに、あるユニット2から他のユニット2へスイッチ40を介して発行（転送）されるコマンドがある。そのようなコマンドとして、ホストプロセッサ1からのリードコマンドおよびライトコマンドの要求を他ユニット2へ転送するためのリードコマンドおよびライトコマンドがある。さらに、ホストプロセッサ1からの上記領域確保コマンドを他ユニット2へ転送するための領域確保要求コマンド、並びに、他ユニット2からの領域確保要求に対する応答としての領域確保成功および領域確保失敗のコマンドがある。

【0046】次に、図1および図4～7を参照して、本実施例のユニット2の構成を詳しく説明する。

【0047】図1は、本実施例の仮想ディスクシステムの1つのユニット2の内部構成を示す図である。

【0048】ユニット2には、6台のSCSIインターフェースのドライブ12から構成されるサブユニット11が複数実装されている。なお、このサブユニット11の数やSCSIドライブ12の数は、本発明の効果を得るには、特に制限は無い。

【0049】ユニット2は、さらに、チャンネルバスディレクタ5、2個のクラスタ13-1、13-2、およびバッテリバックアップ等により不揮発化された半導体メモリであるキャッシュメモリ7を備えている。このキャッシュメモリ7には、制御用のテーブルが格納されている。このテーブルについては、後に詳述する。キャッシュメモリ7およびその中のテーブルは、二つのクラスタ13-1、13-2から共有で使用される。

【0050】クラスタ13（13-1、13-2）は独立に動作可能なバスの集合で、各クラスタ13間においては電源、および回路は全く独立となっている。クラスタ13-1は、2個のチャンネルバス6-1、6-2、2個のドライブバス8-1、8-2、およびスイッチバス10から構成されている。クラスタ13-2の構成も同じである。

【0051】チャンネルバス6（6-1、6-2）は、チャンネル4とキャッシュメモリ7との間のバスである。ドライブバス8（8-1、8-2）は、キャッシュメモリ7とドライブ12との間のバスである。スイッチバス10は、キャッシュメモリ7とスイッチ40との間のバスである。チャンネルバス6、ドライブバス8、およびスイッチバス10は、キャッシュメモリ7を介して接続され

ている。ホストプロセッサ1より発行されたコマンドは、外部インターフェースバス4を通じてチャネルバスディレクタ5に発行される。

【0052】2個のクラスタ13-1、13-2は、それぞれ、2個のバスで構成される。したがって、1つのユニット2は、合計4個のバスを備えている。このことから、ユニット2ではホストプロセッサ1からのコマンドを同時に4個まで受け付けることが可能である。

【0053】図4は、図1のクラスタ13の内部構造を示した図である。なお、図において、共通のものは*-1、*-2のように同じ番号の後に-1や-2を付けて示しているが、以下の説明ではこれらの-1、-2などを除いた番号でそのものを代表させる。例えば、MP20は、MP20-1およびMP20-2の両者を代表する。

【0054】チャネルバスディレクタ5は、各チャネル4に接続されたインターフェースアダプタ(IF Adp)15、およびチャネルバススイッチ16を備えている。チャネルバス6は、チャネルインタフェース(CH IF)21、データ制御回路(DCC)22、チャネル側キャッシュアダプタ(C Adp)23、およびマイクロプロセッサ(MP)20を備えている。チャネルインタフェース21は、データ線18によりチャネルバススイッチ16と接続されている。

【0055】キャッシュメモリ7には、仮想ドライブ管理テーブル31、ユニット管理テーブル32、および物理ドライブ管理テーブル33が設けられている。ドライブバス8は、ドライブ側キャッシュアダプタ(C Adp)26、およびドライブインタフェース回路(Drive IF)27を備えている。スイッチバス10は、スイッチインタフェース回路(SW IF)25、データ制御回路(DCC)28、およびスイッチ側キャッシュアダプタ(C Adp)24を備えている。

【0056】9はドライブユニットバスである。17は、チャネルバススイッチ16、チャネルインタフェース17、データ制御回路22、チャネル側キャッシュアダプタ23、マイクロプロセッサ20、キャッシュメモリ7、ドライブ側キャッシュアダプタ26、ドライブインタフェース回路27、スイッチインタフェース回路25、データ制御回路28、およびスイッチ側キャッシュアダプタ24を接続する制御信号線である。

【0057】ホストプロセッサ1から発行されたコマンドは、インターフェースアダプタ(以下、IF Adp)15により取り込まれ、マイクロプロセッサであるMP20はクラスタ内の外部インターフェースバス4の中で使用可能なバスがあるか否かを調べる。使用可能な外部インターフェースバス4がある場合、MP20は、チャネルバススイッチ16を切り換えて、コマンドの受け付け処理を行なう。コマンドを受け付けることができない場合は、受付不可の応答をホストプロセッサ1へ送

る。

【0058】図5の上段に、キャッシュメモリ7の内部の適当な領域に格納されている仮想ドライブ管理テーブル31を示す。

【0059】上述したように、ホストプロセッサ1は、CKDフォーマット対応の単体ドライブと同じインターフェースでコマンドを発行してくる。これに対し、本実施例の仮想ディスクシステムでは、ホストプロセッサ1が単体と認識しているドライブは、実際は複数のドライブにより構成される仮想的なドライブである。このため、MP20は、ホストプロセッサ1より指定してきた仮想ドライブアドレス(ドライブアドレスとCCHH)を、実際にデータが格納されている(または格納する)物理ドライブに対するドライブアドレスとドライブ内アドレスに変換する必要がある。図5の仮想ドライブテーブル31は、そのような仮想ドライブアドレスから物理ドライブアドレスへの変換を行うためのテーブルである。

【0060】仮想ドライブテーブル31は、ホストプロセッサ1から指定される仮想ドライブのドライブアドレス51、およびその仮想ドライブ内アドレスであるCCHH52のフィールドを備えている。また、仮想ドライブアドレスに対応する物理ドライブのドライブアドレス53、およびそのドライブ内アドレスであるCCHH54のフィールドを備えている。

【0061】さらに、この仮想ドライブアドレスのデータがキャッシュメモリ7内に存在する場合の、そのデータのキャッシュメモリ7内のアドレスを格納するキャッシュアドレス55のフィールド、およびキャッシュメモリ7内にデータを保持している場合にオン(1)が登録されるキャッシュフラグ56のフィールドがある。

【0062】図5の下段は、1つのユニット内の物理ドライブに実際にデータが格納されている様子を示す図である。DR1、DR2、…、DR6は、このユニット内の物理ドライブのドライブアドレスである。図5の上段の仮想ドライブ管理テーブル31では、仮想ドライブアドレス51がVDR1でCCHH52がADR1のデータが、物理ドライブアドレス53がDR1でCCHH54がPADR1の位置に格納されていることが分かる。これに対応して図5の下段では、物理ドライブアドレスがDR1でCCHHがPADR1の位置に、仮想ドライブアドレスがVDR1でCCHHがADR1のデータVDR1-1が格納されている。他のデータについても、同様に、仮想ドライブ管理テーブル31に例示した通りの位置に格納されている。

【0063】仮想ドライブ管理テーブル31の物理ドライブアドレス53にUN2あるいはUN3と記載され、CCHHが空欄(ー)のものがあるが、これは当該仮想ドライブアドレスのデータが他ユニットにあることを示している。UN2あるいはUN3は、ユニットを特定す

るユニット番号である。

【0064】例えば、仮想ドライブアドレス51がVDR1でCCHH52がADR3のデータはユニット番号UN2である他ユニットにあり、仮想ドライブアドレス51がVDR2でCCHH52がADR1のデータはユニット番号UN3である他ユニットにあることが分かる。他ユニット内でどの物理ドライブアドレスにデータがあるかは、他ユニットの仮想ドライブ管理テーブル31に記載されている。

【0065】図6は、キャッシュメモリ7の内部の適当な領域に格納されている物理ドライブ管理テーブル33である。

【0066】物理ドライブ管理テーブル33は、そのユニット2にある物理ドライブの空き領域を管理するためのテーブルである。物理ドライブ管理テーブル33は、そのユニット2にある物理ドライブのドライブアドレス71のフィールド、およびドライブ内アドレスであるCCHHのフィールド72を有する。

【0067】また、その物理ドライブアドレス71とCCHH72とにより指定される物理ドライブのトラック(CCHH)毎に、そのトラックが既に仮想ドライブの領域として割当てられている場合にオン(1)が登録される仮想ドライブ割当てフラグ73のフィールドを有する。さらに、物理ドライブ毎に、仮想ドライブの領域として割当てられていないトラック、すなわち空きトラックの数を保持するフィールド74を有する。

【0068】図7は、キャッシュメモリ7の内部の適当な領域に格納されているユニット管理テーブル32である。

【0069】ユニット管理テーブル32は、ユニット番号からスイッチアドレスを求めるためのテーブルである。61がユニット番号のフィールド、62がそのユニット番号61に対応するスイッチアドレスのフィールドである。ユニット番号は、各ユニットに固有の番号である。スイッチアドレスは、スイッチ40を介して当該ユニットにコマンドを発行するときに指定するアドレスである。

【0070】これらの、テーブル31、32、33は、システムの電源をオンしたときに、MP20により、特定のドライブ12からキャッシュメモリ7に自動的に読み込まれる。一方、電源をオフするときは、MP20により、キャッシュメモリ7内のそれぞれのテーブルが元のドライブ内領域に自動的に格納される。

【0071】次に、本実施例の仮想ディスクシステムの具体的なI/O処理について詳しく説明する。具体的には、あるユニット2で外部から(ホストプロセッサ1または他のユニット2から)各種のコマンドを受けたときにMP20が実行する処理について説明する。以下では、①ホストからのリード/ライト処理、②他ユニットからのリード/ライト処理、③ホストからの仮想ドライ

ブ作成処理、④ホストからの領域確保処理、⑤他ユニットからの領域確保処理の順に説明する。

【0072】①まず、ホストプロセッサ1から、すなわちバス4を介してリード/ライト(読み込み/書き込み)コマンドを受取った場合の処理の全体の流れを説明する。

【0073】図8は、ホストプロセッサ1からリード/ライトコマンドを受取った場合のMP20の処理の流れを示す。ホストプロセッサ1からコマンドを受取ると、MP20は、仮想ドライブ管理テーブル31(図5)を参照して、アクセスの対象となる物理ドライブアドレスを求める(ステップ101)。次に、その対象となる物理ドライブが、自ユニット内のものか他ユニット内のものかを判別する(ステップ102)。

【0074】対象となる物理ドライブ12が自ユニットにある場合は、自ユニットにおいてリード/ライト処理を行う(ステップ103)。対象となる物理ドライブ12が他ユニットにある場合は、仮想ドライブ管理テーブル31(図5)の物理ドライブアドレス53のフィールドに書かれているユニット番号からユニット管理テーブル32(図7)を参照して対象ユニットのスイッチアドレス54を求める(ステップ104、105)。そして、スイッチバス10を介して、求めたスイッチアドレスの対象ユニットへ、リード/ライトコマンドを発行(転送)するよう指示する(ステップ106)。他ユニットへリード/ライトコマンドを発行するときは、ホストプロセッサ1から受けたオペランドも含めて他ユニットへ転送する。

【0075】自ユニットにおけるリード/ライトコマンドの処理について、図4を用いて、詳細に説明する。以下の説明は、上述した図8の処理の全体の流れのうち、ステップ101→ステップ102→ステップ103の処理を詳しく説明するものである。これは、自ユニットにアクセス対象の領域がある場合であり、従来のディスクシステムと同様の動作である。

【0076】図4を参照して、ホストプロセッサ1より発行されたコマンドは、IF Adp15を介して取り込まれ、MP20によりリード(読み出し)要求かライト(書き込み)要求か解読される。

【0077】まず、読み出し要求の場合の処理方法を以下に示す。MP20が読み出し要求のコマンドを認識すると、MP20は送られてきた仮想ドライブアドレスから仮想ドライブ管理テーブル31(図5)を参照し、当該データの物理ドライブアドレスへの変換を行ない、さらにそのデータがキャッシュメモリ7内に存在するかどうかをキャッシュフラグ56を調べて判定する。

【0078】キャッシュフラグ56がオンでキャッシュメモリ7内に格納されている場合(キャッシュヒット)は、MP20が、キャッシュメモリ7から当該データを読み出す制御を開始する。一方、キャッシュメモリ7内

に当該データが無い場合（キャッシュミス）は、当該ドライブ12へその内部の当該データを読み出す制御を開始する。

【0079】キャッシュヒット時の処理について説明する。MP20は、求めたキャッシュアドレス55を用いて、キャッシュメモリ7へ当該データを読み出しに行く。具体的には、MP20の指示の元で、キャッシュアダプタ回路（C Adp）23によりキャッシュメモリ7から当該データは読み出される。C Adp 23は、キャッシュメモリ7に対するデータの読み出しおよび書き込みをMP20の指示で行う回路であり、キャッシュメモリ7の状態の監視、各読み出し、書き込み要求に対し排他制御を行う回路である。

【0080】C Adp 23により読み出されたデータは、データ制御回路（DCC）22の制御によりチャネルインターフェース回路（CH IF）21に転送される。CH IF 21では、チャネルインターフェースの protocols に変換し、チャネルインターフェースに対応する速度に速度調整する。具体的には、ホストプロセッサ1とユニット2との間のチャネルインターフェースを光のインターフェース（光チャネル）にした場合、光のインターフェースの protocols を、ユニット2内では電気処理での protocols に変換する。CH IF 21における protocols 変換および速度調整後は、チャネルバスディレクタ5において、チャネルバススイッチ16が外部インターフェースバス4を選択しIF Adp 15によりホストプロセッサ1へデータ転送を行なう。

【0081】一方、キャッシュミス時は、以下のように処理される。MP20は、Drive IF 27に対し、当該ドライブ12への読み出し要求を発行するように指示する。Drive IF 27では、SCSI の読み出し処理手順に従って、読み出しコマンドをドライブユニットバス9-1または9-2を介して発行する。

【0082】Drive IF 27から読み出しコマンドを発行された当該ドライブ12においては、指示されたドライブ内アドレスへ読み書きヘッドをシークし、回転待ちのアクセス処理を行なう。当該ドライブ12におけるアクセス処理が完了した後、当該ドライブ12は当該データを読み出し、ドライブユニットバス9を介してDrive IF 27へ転送する。

【0083】Drive IF 27では、転送されてきた当該データをドライブ側のキャッシュアダプタ回路（C Adp）26に転送し、C Adp 26ではキャッシュメモリ7にデータを格納する。このとき、C Adp 26は、キャッシュメモリ7にデータを格納することをMP20に報告する。MP20は、この報告に基づいて、仮想ドライブ管理テーブル31（図5）の当該エントリのキャッシュフラグ56をオン（1）にし、キャッシュアドレス55にキャッシュメモリ7内のデータを格納したアドレスを登録する。その後は、キャッシュヒ

ット時と同様な手順でホストプロセッサ1へ当該データを転送する。

【0084】一方、書き込み時は以下のように処理される。まず、MP20は、ホストプロセッサ1から書き込み要求のコマンドを受け取った後、コマンドを受け取ったMP20が所属するクラスタ13内の各チャネルバス6において処理可能かどうかを調べ、可能な場合は処理可能だという応答をホストプロセッサ1へ返す。

【0085】ホストプロセッサ1では、処理可能だという応答を受け取った後に、当該ユニット2へデータを転送する。このとき、ユニット2では、MP20の指示により、チャネルバスディレクタ5のチャネルバススイッチ16が当該外部インターフェースバス4とIF Adp 15を当該チャネルバス6と接続し、ホストプロセッサ1とユニット2との間の接続を確立する。ホストプロセッサ1とユニット2との間の接続を確立した後、ホストプロセッサ1からのデータ転送を受け付ける。

【0086】ホストプロセッサ1から転送されてきた書き込みデータは、MP20の指示により、CH IF 21による protocols 変換が施され、外部インターフェースバス4での転送速度からユニット2内での処理速度に速度調整される。CH IF 21における protocols 変換および速度制御の完了後、データはDCC 22によるデータ転送制御を受け、C Adp 23に転送され、C Adp 23によりキャッシュメモリ7内に格納される。

【0087】このとき、読み出しと同様に仮想ドライブ管理テーブル31（図5）によりアドレス変換を行い、物理デバイスアドレスを求める。また、ホストプロセッサ1から送られてきた情報が書き込みデータの場合は、キャッシュメモリ7に格納したアドレスをキャッシュアドレス55に登録する。このとき、書き込みデータをキャッシュメモリ7内に保持するときは、キャッシュフラグ56をオン（1）とし、保持しない場合はオフ（0）とする。

【0088】このようにキャッシュメモリ7に新データを格納したのをMP20が確認したら、MP20は、書き込み処理の完了報告をホストプロセッサ1に対しチャネル4を介して報告する。

【0089】なお、キャッシュメモリ7内に保持されている新データに対し、さらに書き込み要求がホストプロセッサ1から発行された場合は、キャッシュメモリ7内に保持されている新データを書き替える。その後、Drive IF 27に対し、各々の当該ドライブ12に対し書き込み処理を行なうように指示する。

【0090】ドライブ12への書き込みが完了すると、当該ドライブ12はDrive IF 27に完了報告を行ない、Drive IF 27がこの完了報告を受け取ったことを、MP20に報告する。このとき、MP20は、この新データをキャッシュメモリ7上に残さない場

合は、この報告を元にテーブル31のキャッシュフラグ56をオフ(0)にする。キャッシュメモリ7内に書き込む新データを保持する場合は、書き込み後のキャッシュアドレス55に、キャッシュメモリ7内の新データが格納されているアドレスを登録し、キャッシュフラグ56をオン(1)とする。

【0091】②次に、他ユニットからのリード/ライト処理の要求を受取ったときの処理の流れを説明する。これは、図8のステップ106で発行されたリード/ライト要求を受け取ったユニット2における処理である。

【0092】図9に、他ユニット2からのリード/ライトコマンドをスイッチバス10から受取ったときの処理の流れを示す。

【0093】まず、MP20は、仮想ドライブ管理テーブル31(図5)を参照して、アクセスの対象となる物理ドライブアドレスを求める(ステップ111)。次に、スイッチバス10から受けたリード/ライトコマンドを、通常のリード/ライトコマンドと同様に処理する(ステップ122)。

【0094】この図9の処理の詳細は、上記①で図4を参照して説明した処理手順とほぼ同じである。ただし、上記①ではホストプロセッサ1から発行されたコマンドがバス4を介してMP20に入力するが、図9の処理では他ユニット2からスイッチバス10を介してMP20に入力する。

【0095】また、他ユニット2からのコマンドが読み出し要求であったときの処理手順は、上記①で図4を参照して説明した読み出し要求の場合の処理方法と同じである。本実施例のホストプロセッサ1は要求を発行したチャンネルとその要求に対する応答の割り込みが返ってくるチャンネルとは、異なっているにもかかわらず、他ユニットから転送された読み出し要求コマンドに対して、読み出しデータは直接ホストプロセッサ1に返すようにしている。

【0096】他ユニットからのコマンドが書き込み要求であったときの処理手順も、上記①で図4を参照して説明した書き込み時の処理と同様である。ただし、上記①では書き込みデータはホストプロセッサ1からチャンネルバス6を介して転送されてくるが、図9の処理ではスイッチバス10を介して他ユニットから書き込みデータが転送されてくる。

【0097】③次に、図10を参照して、ホストプロセッサ1から仮想ドライブ作成コマンドを受け取った場合の処理の流れを説明する。ホストプロセッサ1は、仮想ドライブの管理元のユニット番号と仮想ドライブアドレスとをオペランドとして仮想ドライブ作成コマンドを発行する。オペランドで指定された管理元ユニットでは、以下の処理を行う。

【0098】ホストプロセッサ1から仮想ドライブ作成コマンドを受け取ると、MP20は、仮想ドライブ管理

テーブル31(図5)の空きエントリを確保し(ステップ141)、コマンドのオペランドとして受け取った作成すべき仮想ドライブのアドレスをフィールド51に設定する(ステップ142)。

【0099】④次に、図11を参照して、ホストプロセッサ1から領域確保コマンドを受け取った場合の処理の流れを説明する。ホストプロセッサ1は、仮想ドライブアドレスと実際の領域を確保するユニットのユニット番号と物理ドライブアドレスと確保トラック数とをオペランドとして領域確保コマンドを発行する。当該仮想ドライブアドレス(オペランドで指定された仮想ドライブアドレス)の仮想ドライブの管理元のユニット2では、以下の処理を行う。

【0100】ホストプロセッサ1から領域確保コマンドを受け取ると、MP20は、実際に領域を確保するユニットとしてオペランドで指定されたユニットが、自ユニットであるか他ユニットであるかを判別する(ステップ121)。指定ユニットが他ユニットならば、ユニット管理テーブル(図7)を参照して、指定ユニットのスイッチアドレスを求める(ステップ122)。そして、スイッチバス10を介して当該スイッチアドレスのユニットへ領域確保要求コマンドを発行(転送)する(ステップ123)。他ユニット2への領域確保要求コマンド発行においては、ホストプロセッサ1から受けたオペランドも含めて転送する。

【0101】ステップ121で指定ユニットが自ユニットのときは、物理ドライブ管理テーブル33(図6)を参照して、コマンドのオペランドで指定された物理ドライブアドレスのドライブの空きトラック数74を読み出し、その空きトラック数がオペランドで指定された確保トラック数以上であるか否かを判別する(ステップ124)。空きトラック数が、指定された確保トラック数より小さいときは、その確保トラック数分のトラックは確保できないということだから、ホストプロセッサ1に領域確保失敗の割り込みを返す(ステップ125)。

【0102】ステップ124で空きトラック数が指定された確保トラック数以上のときは、物理ドライブ管理テーブル33(図6)のエントリのうち、コマンドのオペランドで指定された物理ドライブアドレスの先頭エントリからサーチし、仮想ドライブ割り当てフラグ73がゼロのエントリを確保トラック数分だけ確保する。そして、確保したエントリの仮想ドライブ割り当てフラグ73を1に設定し、その指定ドライブの空きトラック数74を確保トラック数だけマイナスして更新する(ステップ126)。

【0103】次に、仮想ドライブ管理テーブル31(図5)において、オペランドで指定された仮想ドライブアドレスの空きトラックエントリに、トラックアドレス(CCHH)52とステップ126で確保したトラックの物理ドライブアドレス53とCCHH54とを設定す

る(ステップ127)。そして、ホストプロセッサ1に領域確保成功の割り込み応答を返す(ステップ128)。

【0104】⑤次に、図12を参照して、他ユニット2から領域確保要求または応答のコマンドを受け取った場合の処理について説明する。

【0105】まず、MP20は、受け取ったコマンドが、領域確保要求のコマンドか応答のコマンドかを判別する(ステップ161)。領域確保要求は、上記④の図11のステップ123で発行されるコマンドである。受け取ったコマンドが他ユニット2からの領域確保要求である場合は、ステップ166~170でそのような領域確保要求を受け取った側のユニット2における処理を行う。応答のコマンドとは、領域確保要求を他ユニット2に出し、それを受けたユニットが領域確保処理を行った結果を知らせる応答である。応答のコマンドを受けたユニット2における処理は、ステップ162~165である。

【0106】ステップ161で受け取ったコマンドが領域確保要求であるときに実行するステップ166~170の処理は、上記④の図11のステップ124~128とほぼ同じである。ただし、図11のステップ125および128ではホストプロセッサ1に領域確保失敗あるいは成功の割り込みを返しているが、図12のステップ167および170ではスイッチパス10を介して要求発行元のユニット2に応答を返すようにしている点が異なる。

【0107】ステップ161で受け取ったコマンドが領域確保に対する応答であるときは、領域確保の成功の応答かあるいは失敗の応答かを判別する(ステップ162)。領域確保失敗の応答ならば、ホストプロセッサ1に領域確保失敗の割り込みを返す(ステップ163)。

【0108】ステップ162で領域確保成功の応答ならば、仮想ドライブ管理テーブル31(図5)の該当する仮想ドライブの物理ドライブアドレス53に、応答を送ったユニットの番号を設定し(ステップ164)、ホストプロセッサ1に領域確保成功の割り込み応答を返す(ステップ165)。

【0109】上記実施例によれば、仮想ドライブ管理テーブル31(図5)により仮想ドライブアドレスと物理ドライブアドレスとの対応を取っているため、上位装置からは従来ドライブを同じインターフェースでアクセスできる仮想ドライブが見えるようにできる。仮想ドライブは実際は複数の物理ドライブ上に構成されており、複数の物理ドライブの統合的な管理が実現される。さらに、ユニットを増設して仮想ドライブ作成コマンドや領域確保コマンドを用いて、容易に仮想ドライブの作成および領域の確保を行うことができる。

【0110】図12は、上記実施例の一変形例を示す。上記実施例では、図2に示したように、ユニット群2は

チャンネル4によりホストプロセッサ1と接続されると同時に、スイッチ40とスイッチ線3によっても接続されている。そして、ユニット2間のコマンドは、スイッチ40を介して転送される構成となっている。

【0111】図12は、その一変形例を示すもので、ユニット2間のコマンド転送をチャンネル4とチャンネルディレクタ45により行うものである。チャンネルディレクタ45は、IBM社のES/9000シリーズ(商品名)のチャンネルディレクタ装置のような公知の技術を用いたものである。

【0112】この変形例では、ユニット間2を接続するスイッチという特別なハードウェアが不要であり、すべてチャンネル4でコマンド転送が可能となり、ユニット2の構成を単純化でき、コストも少なくできるという利点がある。

【0113】なお、上記実施例では、磁気ディスクドライブを用いた例を説明したが、磁気ディスクドライブに限らず、光ディスク装置など他の記憶装置を用いてもよい。

【0114】

【発明の効果】本発明によれば、多数の物理ドライブが実装されていても、それらを少数の仮想ディスクボリュームとして、ホストプロセッサからアクセスすることができるので、ドライブの統合的な管理が実現でき、システム管理者のファイル管理が容易となる。また、ホストプロセッサからは従来と互換性のある仮想ディスクボリュームのみが見えており、従来と同様の方法で仮想ディスクボリュームにアクセスできるので、ソフトウェアの変更が不要である。

【0115】また、ドライブを増設した場合は、既存の仮想ディスクボリュームの容量拡張とすることができるので、新たなディスクボリュームのシステムへの追加やファイルの新ディスクボリュームへの移動作業が不要であり、増設が容易となる。また、統合管理の処理はすべてディスクシステム側で行なうのでホストプロセッサの負荷とならない。さらに、ディスクシステムの複数のユニットで処理を分担できるので、特定の制御装置に負荷が集中せず、高性能なディスクシステムを実現することができる。

【図面の簡単な説明】

【図1】実施例の仮想ディスクシステムのユニットの構成図である。

【図2】仮想ドライブの概念図である。

【図3】実施例の仮想ディスクシステムの構成図である。

【図4】実施例のユニットの詳細構成図である。

【図5】実施例の仮想ドライブ管理テーブルの構造図である。

【図6】実施例の物理ドライブ管理テーブルの構造図である。

【図 7】実施例のユニット管理テーブルの構造図である。

【図 8】実施例のホストプロセッサからのリード/ライト処理のフローチャートである。

【図 9】実施例の他ユニットからのリード/ライト処理のフローチャートである。

【図 10】実施例のホストプロセッサからの仮想ドライブ作成処理のフローチャートである。

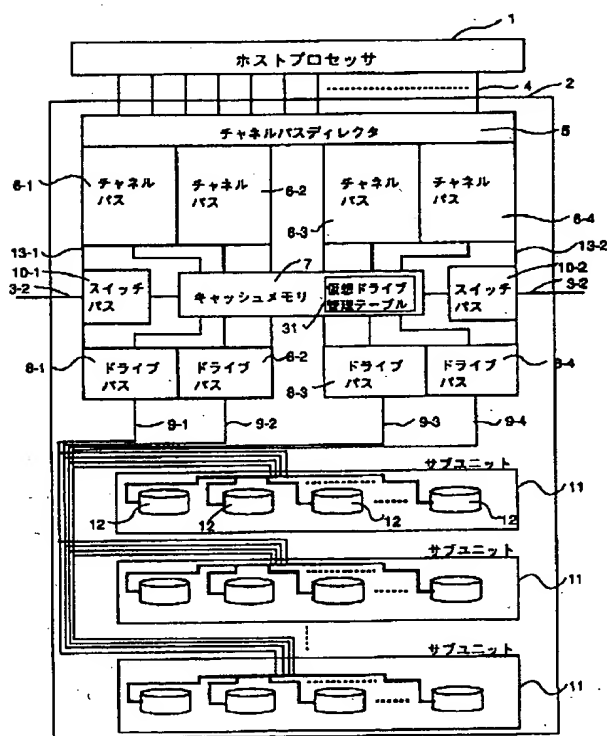
【図 11】実施例のホストプロセッサからの領域確保処理のフローチャートである。

【図 12】実施例の他ユニットからの領域確保処理のフローチャートである。

【図 13】変形例の全体構成図である。

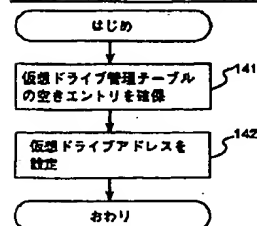
【符号の説明】

【図 1】

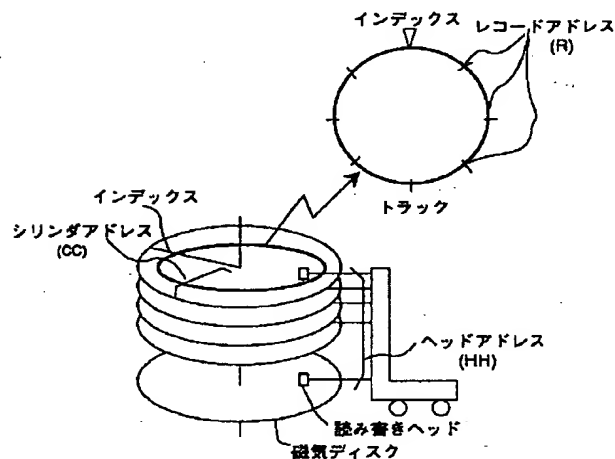


【図 10】

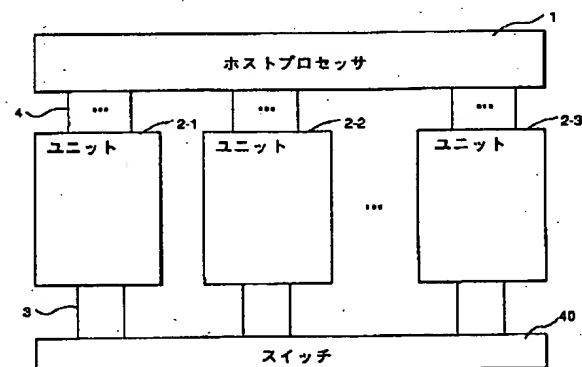
ホストからの仮想ドライブ作成処理



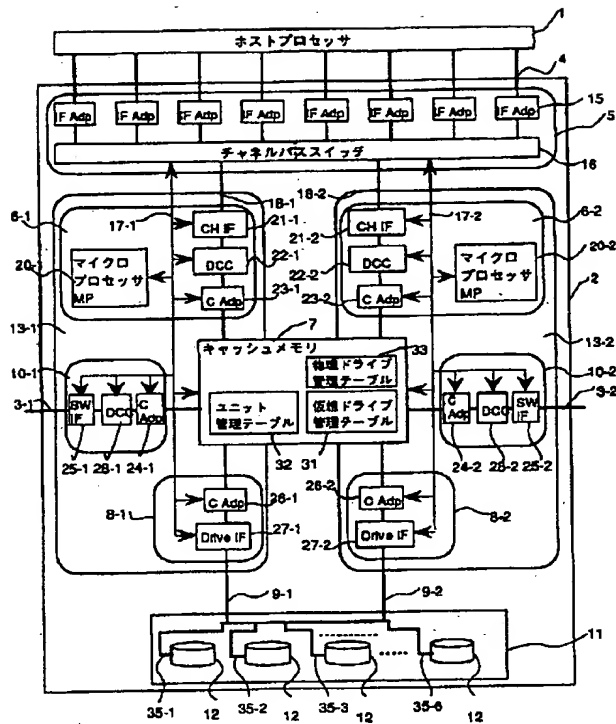
【図 2】



【図 3】

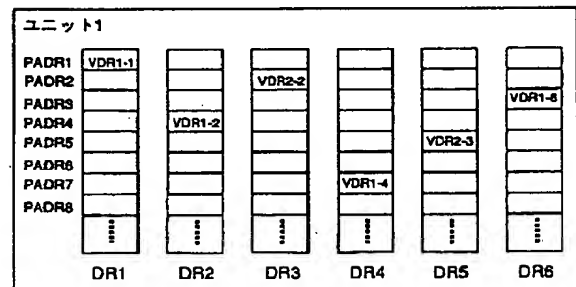


【図 4】



【図 5】

仮想ドライブ アドレス	物理ドライブ アドレス	仮想ドライブ アドレス	物理ドライブ アドレス	キャッシュ アドレス	Cache Flag
ADR1	DR1	PADR1	—	—	0
ADR2	DR2	PADR4	CADR1	1	1
ADR3	UN2	—	—	—	0
...
ADR1	UN3	—	—	—	0
ADR2	DR3	PADR2	—	—	0
ADR3	DR5	PADR5	—	—	0
...



【図 6】

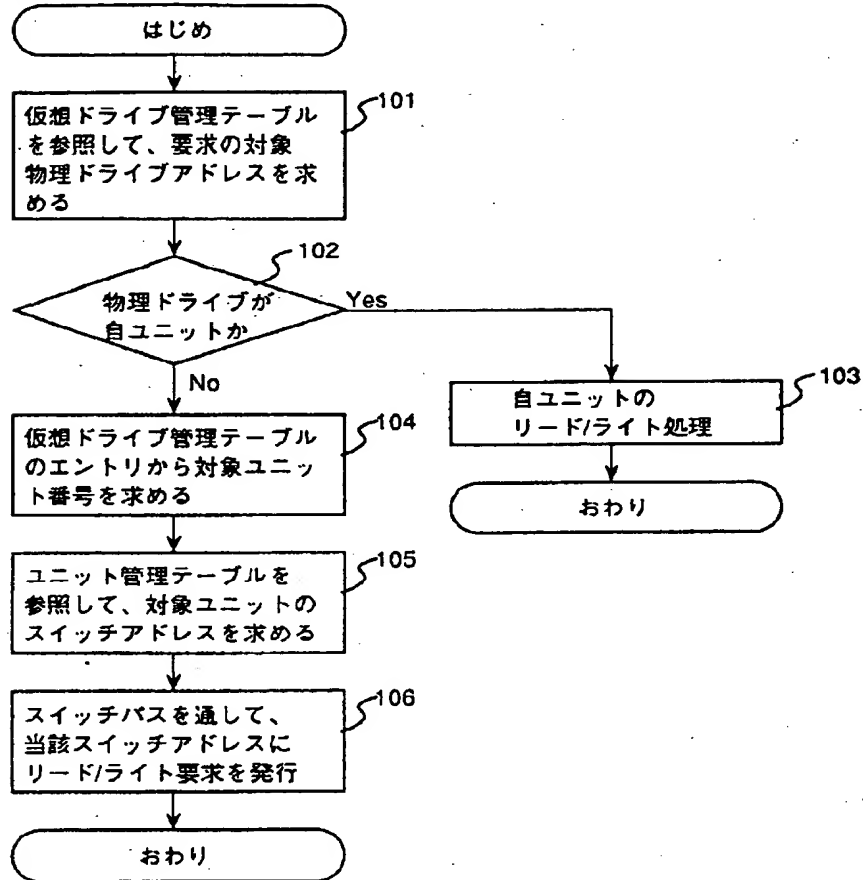
物理ドライブ アドレス	仮想ドライブ 割り当て フラグ	空きトラック数
ADR1	1	256
ADR2	1	
ADR3	0	
...	...	
ADR1	0	1024
ADR2	1	
ADR3	0	
...	...	
...

【図 7】

ユニット番号	スイッチアドレス
UN1	SADR1
UN2	SADR2
...	...

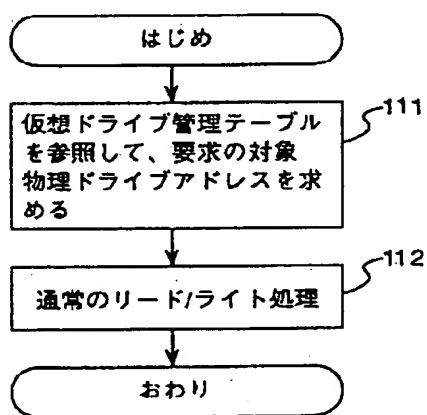
【図8】

ホストからのリード/ライト処理

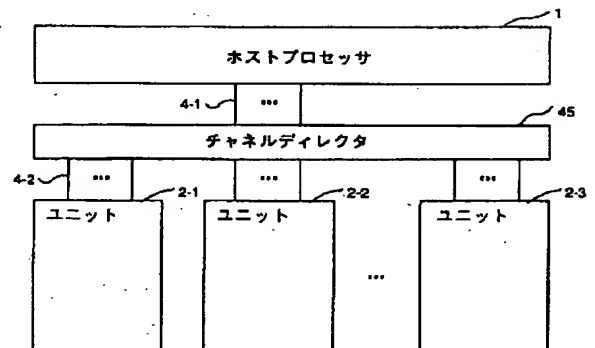


【図9】

他ユニットからのリード/ライト処理

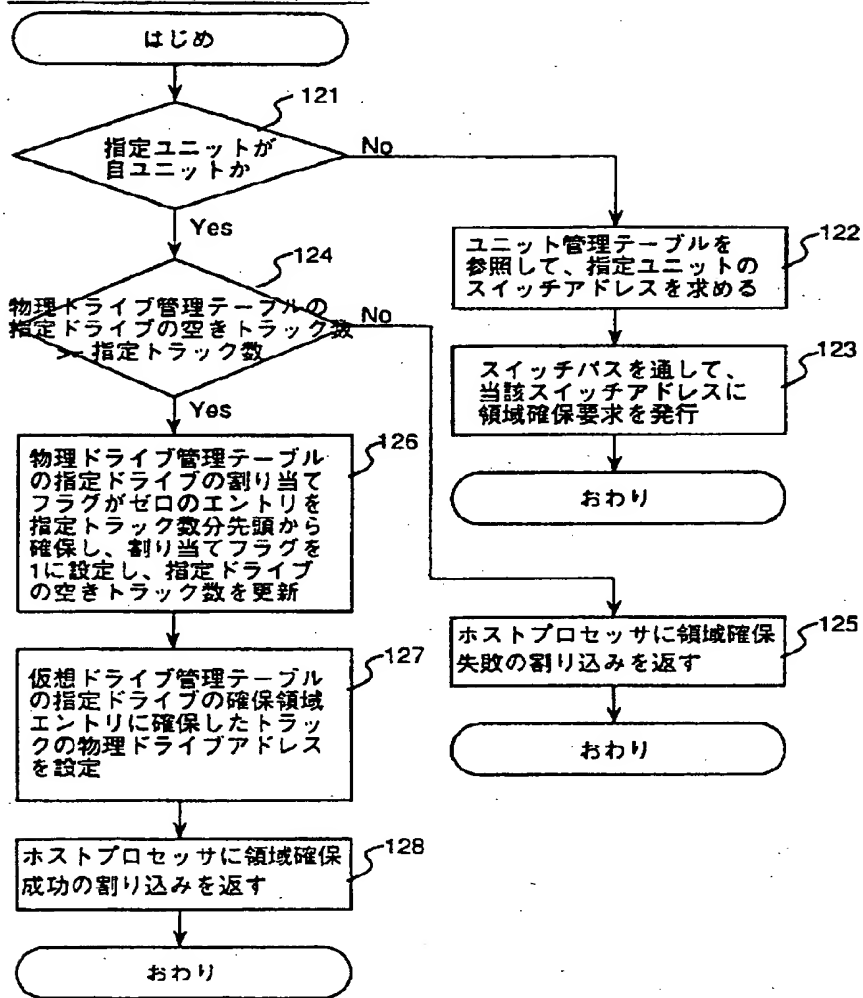


【図13】



【図11】

ホストからの領域確保処理



【図12】

他ユニットからの領域確保処理

